Applications using R

Analysis of categorical variables

Define the categorical variables *cut* and *color*

```
cut=diamonds$cut
color=diamonds$color
```

Display the categories of the categorical variable *cut*

```
unique(cut)
```

Display the absolute frequencies of the categorical variable *cut* and identify the mode

```
table(cut)
```

Draw a bar plot of the categorical variable *cut*

```
plot(cut)
```

or

```
a=ggplot(diamonds, aes(cut))
b=a+geom_bar()
b
```

Display the percentages of the categorical variable *cut*

```
100*table(cut)/length(cut)
```

or

```
100*prop.table(table(cut))
```

Draw the pie chart of the variable *cut*

```
pie(table(cut))
```

Draw a bar plot of the categorical variables *cut* and *color*

```
plot(cut,color)
```

or

```
a=ggplot(diamonds, aes(cut,color))
b=a+geom_count()
```

Run the chi-square test to study the association between two categorical variables, *cut* and *color*

Null hypothesis $H_0: \chi^2 = 0$ (independence, no association)

Alternative hypothesis $H_1: \chi^2 > 0$ (dependence, association)

The decision of the test depends on the *p*-value.

If *p*-value < 0.05, there is an evidence against H_0 (we reject H_0).

If *p*-value $\geq$ 0.05, there is an evidence in favor of H_0 (we do not reject H_0)

```
chisq.test(cut,color)
```

Analysis of numerical variables

Install and load the package *ggplot2*

Consider the dataset *diamonds*

View the dataset *diamonds*

```
View(diamonds)
```

Define the variable *carat* from the dataset *diamonds*

```
carat=diamonds$carat
```

Compute the mean, the median, the minimum, and the maximum of the variable *carat*

```
m=mean(carat)
Me=median(carat)
min(carat)
max(carat)
```

Compute the first and third quartile of the variable *carat*

```
Q1=quantile(carat,0.25)
Q3=quantile(carat,0.75)
```

Compute the p-quantile of the variable *carat*

```
Qp=quantile(carat,p)
```

Compute the frequency of the value 0.21 for the variable *carat*

```
length(carat[carat==0.21])
```

Compute minimum value, maximum value, mean, median, first and third quartile of the variable *carat*

```
summary(carat)
```

Draw the histogram of the variable *carat*

```
hist(carat)
```

or

```
a=ggplot(diamonds, aes(x=carat))
b=a+geom_histogram()
b
```

Draw the histogram of the variable *carat* with 50 bins

```
hist(carat,50)
```

or

```
a=ggplot(diamonds, aes(x=carat))
b=a+geom_histogram(bins=50)
b
```

Draw the histogram of the variable *carat* with bins equal to 0.10

```
a=ggplot(diamonds, aes(x=carat))
b=a+geom_histogram(binwidth=0.10)
b
```

Draw the histogram (with densities and 50 bins) and the density plot (in blue) of the variable *carat*

```
hist(carat,50,freq=F)
lines(density(carat),col=”blue”,lwd=2)
```

or

```
a=ggplot(diamonds, aes(carat))
b=a+geom_histogram(bins=50, aes(y = after_stat(density)))
c=b+geom_density(col=”blue”)
c
```

Draw a red and green histogram (with densities and 50 bins) and the density plot (in blue) of the variable *carat*

```
hist(carat, 50, freq=F, col=”green”)
lines(density(carat), col=”blue”, lwd=2)
```

or

```
a=ggplot(diamonds, aes(x=carat))
b=a+geom_histogram(bins=50, color="red", fill="green",  aes(y = after_
stat(density)))
c=b+geom_density(col="blue")
c
```

Define the variable *price* from the dataset *diamonds*

```
price=diamonds$price
```

Draw the scatterplot between variables *carat* and *price*

```
plot(carat, price)
```

or

```
a=ggplot(diamonds,aes(x=carat, y=price))
b=a+geom_point()
b
```

Draw a smoothed scatterplot between variables *carat* and *price*

```
smoothScatter(carat, price)
```

or

```
a=ggplot(diamonds, aes(x=carat, y=price))
b=a+geom_bin2d(bins=250)
```

Compute the correlation between variables *carat* and *price*

```
cor(carat, price)
```

Run the correlation test to study between variables *cut* and *color*

Null hypothesis $H_0: r = 0$ (no correlation)

Alternative hypothesis $H_1: r \neq 0$ (correlation)

The decision of the test depends on the *p*-value.

If *p*-value < 0.05, there is an evidence against H_0 (we reject H_0).

If *p*-value $\geq$ 0.05, there is an evidence in favor of H_0 (we do not reject H_0)

```
cor.test(carat, price)
```

Draw the scatterplot matrix of the variables *carat*, *price,* and *x* of the dataset *diamonds*

```
x=diamonds$x
pairs(cbind(carat, price, x))
```

or

```
library(GGally)
ggpairs(as.data.frame(cbind(carat, price, x)))
```

Data handling

Check the presence of missing values in a dataset
```
anyNA(diamonds)
```

If the output is FALSE, we have no missing value.
If the output is TRUE, we have one or more missing values, and we can identify them
```
colSums(is.na(diamonds))
```

If you want to delete the rows with NA (if any), we define a new dataset
```
diamonds2=na.omit(diamonds)
```

Check the presence of possible outliers for the variable *carat*
```
boxplot(x)
```

If there are possible outliers they can be listed using
```
outliersx=boxplot(x)$out
```

Regression analysis

Suppose we have imported the dataset *DatasetMarketing* reporting for 170 companies the following variables:
Nationality
Youtube advertising expenses
Facebook advertising expenses
Newspaper advertising expenses
Sales
The aim is the estimate of the best model to predict the sales of a company

First, check for the presence of missing values
```
anyNA(DatasetMarketing)
```

Given the answer (TRUE), we check how many missing values are present.
```
colSums(is.na(DatasetMarketing))
```

Given that there is just one missing value, we can omit the row containing it and define a new dataset
```
DatasetMarketing2=na.omit(DatasetMarketing)
```

Define the variables
```
nationality=DatasetMarketing2$nationality
youtube=DatasetMarketing2$youtube
facebook=DatasetMarketing2$facebook
news=DatasetMarketing2$newspaper
sales=DatasetMarketing2$sales
```

Then, we check for possible outliers and inaccuracies.

Next step is the estimate of the complete regression model
```
mod0=lm(sales ~ nationality+youtube+facebook+news)
```

Estimate the best model and display the output
```
modB=step(mod0)
summary(modB)
```

Make a prediction with the following categories/values:
Nationality=US
Youtube expenses = 300
Facebook expenses = 100
Newspaper expenses = 100
```
newdata=data.frame(nationality=”US”,youtube=300,facebook=100,news=100)
predict(modB,newdata)
```